PATCH DYNAMICS FOR MULTISCALE PROBLEMS

The engineering analysis and microscopic simulations required for predicting materials' properties from atomistic descriptions require new approaches for predicting macroscopic properties. Patch dynamics bridges the gap between the time and space scales at which the microscopic models operate, helping predict system-level behavior.

There are important systems—those that couple molecular dynamics with a material's macroscale behavior or use a Boltzmann particle model to predict large-scale patterns in a fluid flow, for example—that must be modeled on relatively long time or space scales, but the dynamics can be advanced only on short time or space scales. First developed by Yannis Kevrekidis and his collaborators,[1–5] *patch dynamics* is an efficient approach for bridging these scales. Essentially, patch dynamics uses locally averaged properties of short space–time scales to advance and predict long space–time scale dynamics. Inspired by Kevrekidis' early success, E. Weinan and Bjorn Engquist recast the approach into an algorithm they call the *heterogeneous multiscale method*[6] and demonstrated its applicability to additional problems.

In this article, I describe how patch dynamics bridges multiple scales and provides new tools for scientists and engineers trying to predict the dynamics of long space–time scales. When successfully applied, patch dynamics can use microscopic descriptions of a problem to create a system-level framework that helps predict macroscopic properties from direct numerical simulations of relevant microscopic models.

1521-9615/05/$20.00 © 2005 IEEE
Copublished by the IEEE CS and the AIP

JAMES M. HYMAN
Los Alamos National Laboratory

Bridging the Macroscale and Microscale

Most continuum models of reaction and transport processes are derived as conservation laws (mass, species, momentum, and energy) whose average properties are described by partial differential equations (PDEs)—for example, the representation of viscous stresses for Newtonian fluids or mass-action chemical kinetics expressions. However, for a growing class of simulations—including crack propagation, molecular dynamics, Boltzmann kinetic theory models, and modeling the membrane of a living cell—microscale models aren't based on PDEs, but on other physically motivated discrete models. The mechanical properties of deforming materials, such as modeling a material's stress and hardening or predicting defect dynamics as a function of load, often hinge on microscopic transitions that macroscopic-averaged PDE models don't accurately account for.

If we need to predict a system's behavior for macroscopic space–time scales when only the microscopic model is available, the computational cost can be prohibitive—as it will be for the fore-

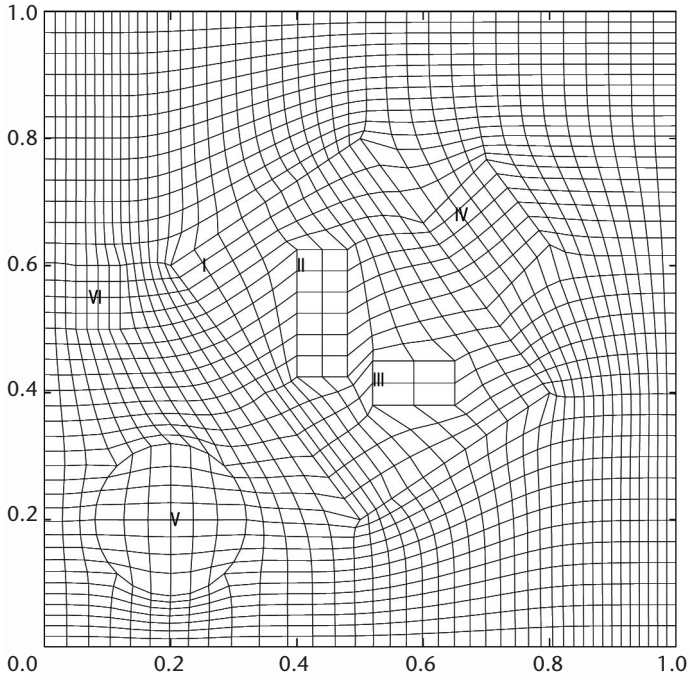

Figure 1. A typical macroscale solution mesh. In patch dynamics and finite difference approximations, the macroscale variables are defined at the grid points of a mesh chosen to resolve the solution. The standard PDE adaptive grid methods can be used to resolve gradients in the macroscale solution. Both patch dynamics and finite difference methods generate time derivatives mesh points; these time derivatives then help advance the solution in time. (Figure courtesy of S. Li.)

seeable future. In situations in which we know a physical process's microscopic description, patch dynamics can help us compute the microscale dynamics on a grid of small patches, which in turn can help us predict macroscale behavior. Patch dynamics circumvents the need for a closed-form macroscopic description of the system and bypasses the need to explicitly define macroscopic equations, but it still delivers macroscopic-level information.

Moreover, macroscale equations might be unavailable because microscale dynamics (such as propagation at the tip of a crack) is a highly nonlinear function of small-scale physics, and continuum models can't capture this singular behavior. In other physical models, the macroscopic equations for average quantities such as mass, momentum, or energy are known, but the equations for the higher moments of the variables' distribution on the microscale aren't. Patch dynamics can predict system behavior of these higher moments for long space–time scales without explicit evolution equations such as PDEs.

Finite difference and finite element methods are standard indispensable algorithms for solving PDEs in science and engineering. The tools and analysis in applying these methods have a lot in common with patch dynamics. In patch dynamics—as well as in finite difference and element methods—we first define a mesh (such as the one in Figure 1) to cover the domain of interest. Next, we define the macroscale solution U at the grid points and approximate the PDEs with finite differences to define the time derivative U_t at the grid points. The finite difference method is an ingenious technique that uses a PDE to combine the neighboring values of the solution on a grid to define a time derivative for the solution at the grid points. These time derivatives advance the solution in time by using a numerical integration method, such as Runge-Kutta. We determine the approach's accuracy by how well the mesh resolves the underlying solution, how accurately the PDE's finite difference approximation defines U_t, and how accurately the numerical integration method advances the solution in time.

As in a finite difference method, the first step in patch dynamics is to define an appropriate grid that resolves the macroscale structure. In finite difference methods, we solve for the value of the averaged microscale at each of the grid points. In patch dynamics, the grid points are stretched into the small patches (regions) where the microscopic model will be solved. Next, we generate microscopic initial conditions in the patch to agree with the macroscale averages at the grid points. The global macroscopic solution is defined by interpolating the macroscale averages at the grid points. This interpolant defines the microscale boundary conditions at the edges of the patches and provides communication across the spatial gaps between the patches. The microscale solution is then advanced a short time in each patch using the microscopic model. The integration of the microscopic model creates changes in the macroscale averages over the patch, thus defining the time derivatives of averaged quantities and their moments. As in the PDE case, we use these time derivatives to advance the macroscopic variables in time via a numerical integration method.

Patch dynamics uses the patches' microscopic models and interpolated boundary conditions in the same way that finite difference methods use a PDE to define time derivatives. Accuracy depends on certain assumptions, such as the macroscale dynamics being separated from the microscale dynamics and the macroscale dynamics being statistically stable to small perturbations in the microscale.

Patch dynamics is related to previous ap-

proaches that use microscopic models and computer analysis to locate bifurcation points or the asymptotic dependence of the control parameter near the bifurcation.[7–10]

Overview of Patch Dynamics

The algorithmic flow for patch dynamics is similar to that of a finite difference code for solving PDEs. Consider Figure 2, the solution of a one-dimensional physical system in which the microscale structure of the solution $u(x)$ varies rapidly, but the locally averaged value of the solution $\langle u(x)\rangle$ varies slowly. The goal of deriving and solving the macroscale equations is to predict the behavior of the solution's smooth average values, without following the microsolution's rapidly varying space–time scales. The patches communicate with each other through the patch boundary conditions, similar to the way finite difference approximations of PDEs communicate the value of the discrete solution at the grid point to the solution at the surrounding grid points. We use the boundary conditions obtained for the macroscale reconstructed solution to define the microscale solution's statistical properties in the buffer region enclosing the patch. The reconstructed macroscale solution might be obtained, for example, by interpolating the macroscopic field variables between the patches with a local piecewise polynomial interpolant.

We can summarize the patch dynamics algorithm for time-dependent problems in the following fashion:

1. Determine the appropriate variables for defining the macroscale solution $U = (\langle u\rangle, \langle u_x\rangle, \langle u_y\rangle, \langle u_{xx}\rangle, \langle u_{xy}\rangle$, and so on). Patch dynamics succeeds only if the slowly varying spatial macroscale quantities are sufficient to determine the system's dynamics. The initial conditions for the macroscale are defined at grid points based on the macroscale length scale ΔX, which adequately samples the macroscale solution. In Figure 2, the grid points x_i and x_{i+1} resolve the macroscale smooth solution before becoming the centers of the patches.
2. Define the patches with length scale Δx at the grid points of the mesh covering the macroscale domain. The microscale dynamics, defined on the small length scale $\Delta x << \Delta X$, will be modeled in each of these patches.
3. Using the macroscopic solution U, define the microscale initial conditions u for each microscale patch to match the average macroscale quantities. In time-dependent problems, the microscale solution from the last time step is adjusted using a defect-correction algorithm so that it has the appropriate averages. This *lifting* procedure is the step in which the macroscale communicates with the microscale.
4. Define the patches' boundary conditions by interpolating the macroscale variables defined at the grid points and then evaluating the interpolant in a buffer region surrounding each patch, as shown in Figure 2. The microscale solution boundary conditions agree statistically with the macroscale interpolant. This is the step in which the patches communicate with each other through the boundary conditions.
5. Using the microscopic model, advance the microscale solution in the patches a short time Δt, as shown in Figure 3.
6. If the current time for the patch solution is sufficient to calculate the statistical averages, then go to 7; else define (lift) U from the microscale and go to 4.
7. Calculate the time derivatives of statistical averages to macroscale solution $U_t = (\langle u\rangle_t, \langle u_x\rangle_t, \langle u_{xx}\rangle_t$, and so on) for the microscale space–time patch shown in Figure 3. This *restriction* procedure is the step in which the mi-

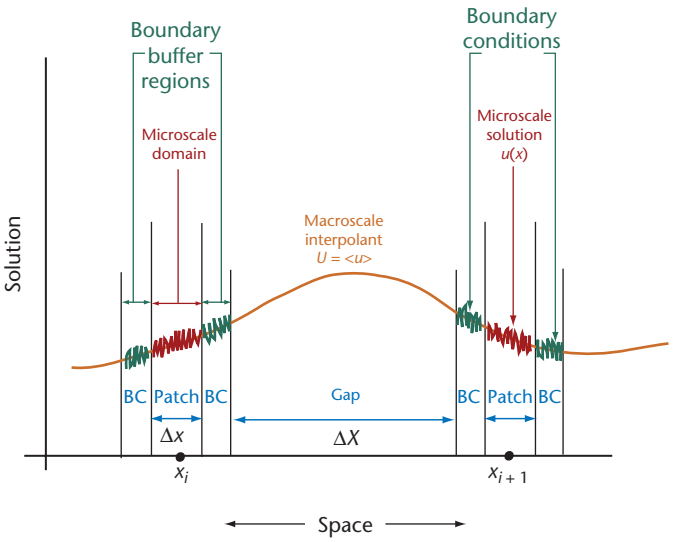

Figure 2. One-dimensional physical system. The microscale variable $u(x)$ varies rapidly, but the macroscopic variables $U = (\langle u\rangle, \langle u_x\rangle, \langle u_{xx}\rangle$, and so on) vary slowly. The boundary conditions for the patches are defined by extending the microscale solution into a buffer region surrounding each patch. The patches communicate with each other via boundary conditions similar to the way finite difference approximations of PDEs communicate among the surrounding grid points.

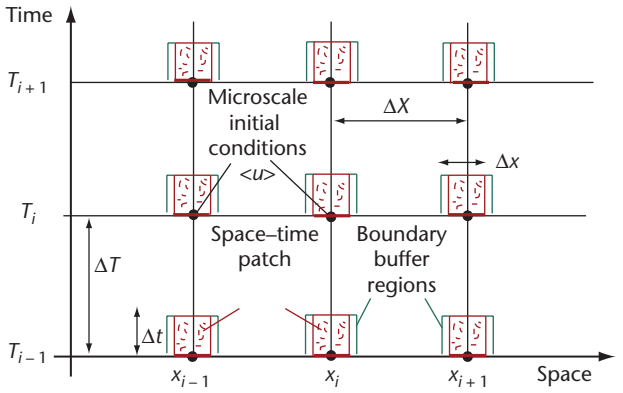

Figure 3. Space–time plot for the patches in Figure 2. The microscale solution is advanced in small space–time patches until we can get accurate approximations of the time derivatives of the macroscale variables *U* over the patch. These time derivatives are used to advance the macroscale variables a macroscale time step, and then the process repeats.

croscale communicates with the macroscopic.

8. Using a standard time integration method, advance the macroscale solution from time T_i to T_{i+1} with macroscale time step $\Delta T >> \Delta t$, as shown in Figure 3.
9. Go to 3.

The computational work of patch dynamics is dominated by solving the microscale model. We can estimate it from the fraction of the space–time domain covered by the patches, such as the one-dimensional space–time plot in Figure 3. If the patches cover 10 percent of the region in each space and time direction, for example, the microscale model is solved on $(0.1)^{d+1}$ of the space–time domain. For this situation in three-dimensional problems ($d = 3$), the microscale model is solved on 0.0001 of the full domain, resulting in a corresponding savings in computer time.

Patch dynamics is not appropriate for all problems. The underlying reference grid must accurately resolve the macroscale solution, and there must be a well-defined microscale model. We also need an appropriate set of macroscale quantities sufficient to define the distribution of microscale variables so that the microscale model is statistically stable (meaning slight variations in microscale initial conditions have little effect on macroscale dynamics). The macroscale spatial and temporal scales must be separated from the microscale.

We can summarize the list of conditions for when patch dynamics is appropriate:

- the macroscale variables are sufficient for defining the microscale initial conditions;
- the microscale dynamics can be sampled sufficiently closely in time and space that macroscale dynamics varies smoothly between sampling locations;
- the macroscale dynamics are separated from the microscale dynamics (that is, $\Delta t << \Delta T$ and $\Delta x << \Delta X$) and the microscale model is statistically stable; and
- the system's dynamics can be advanced on short time (Δt) or spatial (Δx) scales, but the calculation is too expensive to advance the solution on the much longer time ΔT or spatial ΔX scales.

This might seem like a daunting list or restrictions, but a surprising number of problems satisfy all of them, including simulations of reaction-diffusion equations, molecular dynamics, epidemiology, and lattice Boltzmann particle methods.[1,5,7,10] When all these conditions are met, patch dynamics is an appropriate method to try; when they're not, there's always the option of waiting for the next two or three generations of supercomputers before simulating the problem or extending the range of applicability for the method.

Once the microscopic model is fully defined and an appropriate set of macroscale variables is established, we can usually track the success of patch dynamics to three crucial steps: lifting the microscale initial conditions from the macroscale, bridging the spatial gaps, and bridging the temporal gaps. Let's look at each of these steps.[1,3–5]

Lifting the Microscale Conditions

Typically, there are only a few macroscopic mean-field variables of average values, their derivatives, and other statistical moments of the microscopic solution. Many reconstructed solutions can agree with the given spatially averaged statistical properties for the patch, but some of these microscale initial conditions will create exceptionally fast transients in the solution caused by imbalances in the microscale initial conditions made during lifting. These transients must heal quickly for the approach to be effective.[1,3,4]

It's difficult to generate appropriate initial microscale states that retain a physically realistic balance between microsopic forces. Unbalanced initial conditions can generate unacceptable initial transients in the microscale simulation. To minimize this and prevent the introduction of nonphysical artifacts, we use a defect-correction approach and define only the change in the microscale from a previous estimate of the solution (at

a previous time, for example). This approach can be based on the maximum entropy approach, which minimizes the change from the previous solution by explicitly adding an interpolant.

Even the best methods will create microscale initial conditions that generate some transients. If the model is statistically stable, these transients will quickly decay, but their influence on the time-averaged statistical properties, such as $\langle u \rangle_t$ or $\langle u_x \rangle_t$, will decay only linearly in time. In situations where the microscale initial transients take an exceptionally long time to settle down, it can be helpful to predict the time derivatives as well as the macroscopic variables' mixed space–time derivatives. However, defining the time derivatives of the initial conditions to be consistent with both space and time derivatives of U requires solving a small local system of equations. This initialization can be difficult because the average time derivatives are nonlinear nonlocal functions of the microsolutions in several patches.

Regardless of what reconstruction method we've used, however, we've found that the accuracy improved if we applied the same lifting method consistently on every time step.

Bridging the Spatial Gaps

Microscale dynamics requires boundary conditions, which are imposed by defining the microscale solution within a cocoon or buffer region surrounding the patches that agrees with the statistical properties of the macroscale variables. The macroscale variables are defined between the patch grid points by polynomial interpolation. We typically use tensor-product piecewise-Hermite polynomial interpolants for the macroscale variables to define the microscale's statistical properties in the patches' buffer regions. These fast, simple formulas for the interpolation involve only the neighboring patches. In two spatial dimensions, the interpolant, and hence the boundary conditions for the buffer region, are defined from the piecewise polynomials for the quadrant in which the patch boundary resides. The explicit formula for the boundary conditions is a linear combination of the variables defined at the corners of the macroscale cell.

The statistical distribution of the microscale solution in the buffer region is defined to agree with the macroscale interpolant. When the microscale model is a discrete multibody simulation, for example, the microscale solution is extended into the buffer region so that the statistical distribution of the particles' density satisfies the appropriate boundary conditions in the buffer region. In this way, the patches communicate with each other through the interpolated macroscale and boundary conditions.

Because the patches must have statistical well-posed boundary conditions for the microscale, the macroscale interpolant must be converted into a microscale boundary condition appropriate for the microscopic model. If the microscale model were a conservation law in divergence form, the natural boundary conditions would be stated in terms of the fluxes. If we know the underlying microscale fluxes, then they should be included as one of the macroscopic field variables and interpolated between patches to define the boundary conditions. Unfortunately, the fluxes are rarely known, so the boundary conditions are usually defined based on the macroscale solution's value or gradients.

Another consideration for equations with conservation laws is to retain global conservation of the quantities, including the material not being simulated between the patches. If we don't keep track of the material between the patches, the method will be only *approximately* conservative. For the full calculation to preserve global conservation, we must account for the material in the gaps between the patches, which we do by calculating auxiliary macroscale descriptions of the conserved variables within the gaps. The conservative patch dynamics algorithm requires that the conserved variables be updated on every microscale time step by accounting for their flux in and out of neighboring patches. These auxiliary variables play a key role in constructing the macroscale interpolant that defines the boundary conditions. We construct the conservative interplant to agree with the average values of the material in each patch, but the integral of the interpolant between the patches must agree with the material in the gaps.[11]

Bridging the Temporal Gaps

In the numerical solution of PDEs, the equation's time derivative is returned by evaluating an explicit function of discrete values at the grid points. In patch dynamics, we obtain the time derivative through a microscale integration of another model. We then use this microscale integration to obtain the solution's average time derivatives and the time derivatives of the solution's spatial moments.

An effective lifting process will construct the microscale initial conditions for the patches that are in balance so that the solution quickly reaches a balanced state after a fast transient. Even for a statistically stable model, the initial transients caused by the lifting process can create significant errors

in the average properties until the microscale solution settles down. Therefore, calculations for the space- and time-averaged statistical properties should begin after the transients have decayed.

As the microscale solution is advanced, monitoring the rate of change in macroscale variables guides when the initial transients die down and forces are balanced. After the solution has settled down, these transients can be estimated by calculating a running estimate of $\langle u \rangle_t$ (and possibly higher derivatives) during the microsimulation at each microscale time step Δt.

Using the average values of the microscale variables' solution at the current space–time patch, their past values, and their time derivatives, $\langle u \rangle_t$ advances the microscale solution a macroscale time step. The time integration of the patches on the time macroscale can be done with standard time integration methods such as the Adams multistep methods, Runge-Kutta, or even Taylor series methods. Kevrekidis and Gear derived a method especially well suited for patch dynamics called the projected Euler *k-M* method.[2] Their projected Euler *k-M* time-stepper extrapolates the time scales from the microscopic directly to the system level. They take a few small, accurate time steps and then use polynomial extrapolation to use the last few solution values to project the solution ahead in time.

Note that the time derivatives are most accurate if they're defined at the center of the space–time patch. This time isn't at the beginning of the macroscale time step, but slightly further along; the time shift can be accounted for by a slight modification of the standard numerical time integration methods.

Currently, patch dynamics is appropriate only for statistically stable models—that is, small perturbations in the microscale's initial conditions relax (anneal) quickly to a distribution that is relatively independent of the microscale's detailed structure. Microscale statistics must be insensitive to the details of initial conditions. We can empirically test the statistical stability of the initial conditions for a particular model by generating multiple realizations of a model and calculating the relaxation time of the difference between two initial distributions. If the model is found to be sensitive to these small changes in the microscale distribution, patch dynamics isn't an appropriate method

The numerical solution of PDEs is arguably the most powerful computational algorithm for gaining insight into the behavior of complex physical systems. In the past half century, we've built a huge superstructure to support it, including adaptive mesh methods, linear and nonlinear solvers, time integration methods, powerful visualization software, error estimation and analysis tools, sensitivity analysis, and optimization tools.

Unfortunately, PDEs can actually model relatively few phenomena. Substantial time and energy is focused on them because, just as the man who lost his keys spends the night looking under the lamppost (because it's where the light is), computational scientists have spent many long nights studying PDEs—they're where the tools are.

Patch dynamics uses these same tools and can light the way as we search for new approaches beyond the lamppost for investigating the links between micro- and macroscale models. Although solving microscale dynamics for entire problems would be best, the necessary computer resources aren't likely to be available to do so on the foreseeable generations of computers. By bridging the micro- and macroscales, we'll gain a way to systematically create appropriate hybrid models to study the long space–time scales of the macroscale model based on the microphysics.

Patch dynamics is being developed for multiscale, multiphysics models where even the fastest computers are still inadequate to perform truly molecular-scale analysis of complex macroscopic phenomena. The approach has been effective on some simple problems, and researchers are extending and developing it for more complex problems, even as they evaluate the method's effectiveness by comparing it with detailed microsimulations. Patch dynamics is also being combined with adaptive mesh refinement to better resolve solutions when local regions include high gradients. Moreover, patch dynamics is being developed for three-dimensional molecular dynamics and lattice Boltzmann models, heterogeneous hard materials design, biological systems, deformation and failure in hard materials, growth of thin films and nanostructures, and transport and chemical reaction processes in complex fluid flow.[1,5,10,12]

Acknowledgments

The US Department of Energy supported this research through contract W-7405-ENG-36 and the DOE Office of Science Advanced Scientific Computing Research (ASCR) Program in Applied Mathematics Research. The author thanks Yannis Kevrekidis for the countless hours he spent explaining the underlying mathematical foundation of patch dynamics and its potential to

solve a wide range of previously unassailable problems.

References

1. J. Cisternas et al., "Equation-Free Modeling of Evolving Diseases: Coarse-Grained Computations with Individual-Based Models," *Proc. Royal Soc. London A*, vol. 460, no. 2050, 2004, pp. 2701–2779.
2. I.G. Kevrekidis and C.W. Gear, "Projective Methods for Stiff Differential Equations: Problems with Gaps in Their Eigenvalue Spectrum," *SIAM J. Scientific Computing*, vol. 24, no. 4, 2002, pp. 1091–1106.
3. G. Samaey, D. Roose, and I.G. Kevrekidis, "The Gap-Tooth Scheme for Homogenization Problems," to appear in *SIAM Multiscale Modeling and Simulation*, 2005.
4. G. Samaey, I.G. Kevrekidis, and D. Roose, "Patch Dynamics with Buffers for Homogenization Problems," to appear in *J. Computational Physics*, 2005.
5. C. Theodoropoulos et al., "Coarse Bifurcation Studies of Bubble Flow Lattice Boltzmann Simulations," *Chemical Engineering Science*, vol. 59, no. 12, 2004, pp. 2357–2362.
6. E. Weinan, and B. Engquist, "Heterogeneous Multiscale Method," *Comm. Mathematical Science*, vol. 1, 2003, pp. 87–133.
7. L. Chen et al., "From Molecular Dynamics to Coarse Self-Similar Solutions: A Simple Example Using Equation-Free Computation," *US J. Non-Newtonian Fluid Mechanics*, vol. 120, nos. 1–3, 2004, pp. 215–223.
8. C.W. Gear et al., "Equation-Free Coarse-Grained Multiscale Computation: Enabling Microscopic Simulators to Perform System-Level Tasks," *Comm. Mathematical Sciences*, vol. 1, no. 4, 2003, pp. 715–762.
9. O. Runborg, C. Theodoropoulos, and I.G. Kevrekidis, "Effective Bifurcation Analysis: A Time Stepper Based Approach," *Nonlinearity*, vol. 15, 2002, pp. 491–511.
10. C. Theodoropoulos, Y.-H. Qian, and I.G. Kevrekidis, "Coarse Stability and Bifurcation Analysis Using Time-Steppers: A Reaction-Diffusion Example," *Proc. Nat'l Academy of Science*, vol. 97, 2000, pp. 9840–9843.
11. J.M. Hyman, R.J. Knapp, and J.C. Scovel, "High Order Finite Volume Approximations of Differential Operators on Nonuniform Grids," *Physica D*, vol. 60, nos. 1–4, 1992, pp. 112–138.
12. D.I. Kopelevich, A.Z. Panagiotopoulos, and I.G. Kevrekidis, "Coarse-Grained Kinetic Computations for Rare Events: Application to Micelle Formation," *J. Chemical Physics*, vol. 122, no. 4, 2005, pp. 44908.1–44908.13.

James M. Hyman is the group leader of the Mathematical Analysis and Modeling Group at Los Alamos National Laboratory. His technical interests include the numerical solution of partial differential equations and mathematical biology. Hyman has a PhD in mathematics from the Courant Institute of Mathematics at New York University. He is a member of SIAM and the American Mathematical Society. Contact him at hyman@lanl.gov.

Figure courtesy of Matthias Höchsmann, Björn Voss, and Robert Giegerich.